Willis MJ, von-Stosch M.

L0-constrained regression using mixed integer linear programming.

***Chemometrics and intelligent laboratory systems* 2017, 165, 29-37.**

L0-constrained regression using mixed integer linear programming

Mark J. Willis and Moritz von Stosch
School of Chemical Engineering and Advanced Materials
University of Newcastle
Newcastle upon Tyne, NE1 7RU.
mark.willis@ncl.ac.uk, moritz.von-stosch@ncl.ac.uk

Abstract – In this work, sparse regression using a penalized least absolute deviations objective function is considered. Regression model sparsity is promoted using a L_0 - pseudo norm penalty (the cardinality of the model parameter vector). Implemented using mixed integer linear programming (MILP) it is demonstrated that the use of the L_0 - norm (without approximation) enables efficient and accurate solutions to sparse regression problems of practical size. For model development with a large number of potential model parameters (or features) methods to relax the MILP are also developed; using nonlinear function approximations to the L_0- norm, penalty terms are linearized and solved using sequential linear programming. Experimental results (using both simulated and real data) demonstrate that these algorithms are also computationally efficient producing accurate and parsimonious model structures. Applications considered are the development of a calibration model for prediction with Near Infrared (NIR) data and the development of a model for the prediction of chemical toxicity - a quantitative structure activity relationship (QSAR).

Keywords: Sparse regression, mixed integer linear programming

1.0 Introduction

Sparse approximation describes algorithms aimed at minimizing two competing objectives; that an approximation be as accurate and concise as possible. The development, analysis and application of sparse approximation techniques occur in a significant number of scientific and engineering disciplines. Applications include regression, image and signal processing, the identification of mechanistic networks (such as gene regulatory networks) etc. An approach to sparse regression that has gained popularity in recent years uses model parameter regularization; performing regression using the entire set of suggested model input variables and controlling model complexity in order to improve predictive performance, e.g. see [1]. To do this, an objective function is minimized that combines a measure of model prediction error and a term that penalizes complexity, i.e. the term $\|\boldsymbol{\varepsilon}\|_m$ which is the L_m - norm of the error between the measured and predicted model output and the penalty term $P(\hat{\mathbf{b}})$ which is a non-negative function of the estimated model parameters, $\hat{\mathbf{b}} = [\hat{b}_1, \ldots, \hat{b}_r]^{\mathrm{T}}$ (where r is the total number of model parameters),

$$J(\lambda) = \|\varepsilon\|_m + \lambda P(\hat{\mathbf{b}}) \qquad (1)$$

The weighting, λ, is known as the (model) regularization parameter. For increasing values of λ different (sparse) solutions to (1) are obtained (a value of $\lambda = 0$ corresponding to the non-penalized solution (where all parameters will be present in the model) and a value of $\lambda \to \infty$ being a fully penalized solution with all model parameters at zero). The aim is to determine a value of λ, that gives both the best model structure and the associated parameters. This may be achieved using cross-validation strategies using, for example, a second data set or, by the use of the well-known information criteria for model selection, such as the Akaike

information criteria (AIC) proposed by [2] and Bayesian Information Criteria (BIC) proposed by [3].

The Least Absolute Shrinkage and Selection Operator (LASSO) was introduced by [4] to perform regularized regression where the objective is to minimize (1) subject to the penalty (2), the L_1 – norm of the estimated model parameters.

$$P(\hat{\mathbf{b}}) = \left\|\hat{\mathbf{b}}\right\|_1 = \sum_{j=1}^{r} |\hat{b}_j| \tag{2}$$

Since its introduction LASSO has been a popular approach. However, there are a number of known disadvantages of the method, notably that **a)** the parameter estimates obtained using (2) are known to 'shrink' as the regularization weight is increased and as a result LASSO does not always provide the correct model structure, see e.g. [5], **b)** if model input variables are correlated, the regularization process (slowly increasing the value of λ and observing the resulting reduction in model complexity) tends to choose one of the model parameters and ignore the rest of the parameters associated with the input variables that are correlated, **c)** if $r > N$ (where *r* the number of model inputs and *N* is the number of data points) the LASSO is limited to the selection of at most *N* terms in the model, [6], and **d)** if the aim is use the model for the purposes of prediction, because of the parameter shrinkage that occurs during the regularization process, a sub-optimal predictive model will be obtained [7].

A more natural choice for $P(\hat{\mathbf{b}})$ is the pseudo-norm defined as the number of non-zero elements (the cardinality of the parameter vector), subsequently referred to as the L_0 – norm,

$$P(\hat{\mathbf{b}}) = \left\|\hat{\mathbf{b}}\right\|_0 = card\left\{\{\hat{b}_j | \hat{b}_j \neq 0\}\right\} \tag{3}$$

Using the L_0 – norm ensures that in minimizing (1) the most parsimonious model explaining the data is found. However, the L_0 – norm is non-convex and discontinuous and solution of (1) using the L_0 – norm penalty is known to be NP-hard (e.g. when implemented as a best subset regression problem it does not scale to problem sizes where, $r > \sim 30 - 40$). As a result, there are numerous reported works that suggest approximating the L_0 – norm penalty using a continuous smooth function. For example, [8] used (where ϵ_1 is a small number that is used to avoid the discontinuity in the logarithm function at zero),

$$P(\hat{\mathbf{b}}) = \sum_{i=1}^{r} \log(|\hat{b}_j| + \epsilon_1), \varepsilon > 0 \tag{4}$$

The approximation imitates the L_0 – norm in that the penalty term (and hence the objective function) decreases rapidly to zero for small values of $|\hat{b}_j|$ compared to its slower increase for larger values of $|\hat{b}_j|$. Therefore it is better to increase e.g. some of the $|\hat{b}_j|$ while setting others to zero rather than obtaining a compromise solution that contains all coefficients. Other approximations to the L_0 – norm include the use of an exponential function [9], the Smoothly Clipped Absolute Deviation (SCAD) penalty [10], the Bridge Penalty, e.g. see [11], the Seamless L_0 – or SELO, proposed by [12] etc. The idea behind the use of these penalties is the same; they offer a better approximation to the cardinality constraint than the L_1 – norm and therefore the opportunity to more efficiently (in terms of computational overhead) perform sparse regression.

In this paper, as opposed to using an approximation, a MILP implementation of the sparse regression problem that uses the cardinality constraint directly is proposed. This offers a novel regularization strategy, which decouples the parameter estimation and structure identification problems (therefore avoiding parameter estimation bias). Using slack variable constraints the MILP is formulated as a smooth, constrained problem which demonstrates a significant decrease in computational effort required to solve the cardinality constrained MILP for problems where, $r \sim 100 - 150$. For higher dimensional input data novel relaxed versions of the MILP are proposed to further improve algorithm efficiency (in terms of computational effort required to obtain a solution). The relaxed MILPs are implemented via sequential linear programming using an iteratively linearized nonlinear penalty function that approximates the behavior of the L_0 – norm.

2.0 Methods

Given $i = 1, \dots, N$ measured data points a response vector, $\boldsymbol{y}$ (N x 1) and a feature matrix $\boldsymbol{X}$ (N x r) a linear model is defined as (where, $\hat{\mathbf{b}}$ is a vector of model parameters and $\boldsymbol{\varepsilon} = [\varepsilon_{i,1} \quad \cdots \quad \varepsilon_{i,N}]^T$ are random errors with a mean of zero and variance, σ),

$$\boldsymbol{y} = \boldsymbol{X}\hat{\mathbf{b}} + \boldsymbol{\varepsilon}$$

It is assumed that only a small subset of $\{\mathbf{x}_j\}_{j=1}^{r}$ have non-zero parameters and the aim is to efficiently identify this subset and the associated parameters using MILP. As opposed to using a least squares objective function (minimizing the squared error between an output and a predicted output) in this work the measure of prediction error used is the sum of the absolute errors (the L_1 – norm), i.e. $m = 1$ in objective function (1). This provides a popular alternative to Least Squares (L_2 – norm minimization) because it is insensitive to outliers in the data set. Moreover, the L_1 – norm may be formulated as a linear objective function. To ensure regression model sparsity, a set of binary variables (associated with each of the parameters of the model) are used to perform regularization rather than the parameters themselves. The binary variables provide a normalised entropy measure (independent of the magnitude of the regression parameters) and are directly related to the number of parameters in the model (the cardinality of the parameter vector).

2.1 Mixed integer linear programming

The L_1 - norm cost function, including the L_0 - norm regularization penalty is (5).

$$J(\lambda) = \|\boldsymbol{\varepsilon}\|_1 + \lambda\left\|\hat{\mathbf{b}}\right\|_0 \tag{5}$$

Rewriting (5) using a vector of auxiliary variables, $\boldsymbol{z} = (z_1, \dots, z_N)^{\mathrm{T}}$ and binary variables $\boldsymbol{\delta} = (\delta_1, \dots, \delta_r)^{\mathrm{T}}$ (where $\delta_j = 1$ if $\hat{b}_j \neq 0$ and $\delta_j = 0$ if $\hat{b}_j = 0$) gives (6). If this is minimized, subject to the constraints (7) – (10), the MILP implementation is equivalent to (5).

$$J(\lambda) = \sum_{i=1}^{N} z_i + \lambda \sum_{j=1}^{r} \delta_j \tag{6}$$

$$z_i \geq \varepsilon_i \qquad (i = 1, \dots, N) \tag{7}$$
$$z_i \geq -\varepsilon_i$$

$$L_j\delta_j \leq \hat{b}_j \leq U_j\delta_j \qquad (j = 1, \dots, r) \tag{8}$$

$$\delta_j \in \{0,1\} \qquad (j = 1, \dots, r) \tag{9}$$

$$z_i \geq 0 \qquad (i = 1, \dots, N) \tag{10}$$

The decision variables for the MILP are **a)** the z_i $(i = 1, \dots, N)$ where the constraints (7) ensure the smallest possible (positive values) are obtained that minimize (6), **b)** the model parameters $\hat{b}_j$ $(j = 1, \dots, r)$ and **c)** the binary variables, δ_j $(j = 1, \dots, r)$. The L_j and U_j represent the upper and lower bounds on the model parameter values, $\hat{b}_j$. The constraints (8) and (9) ensure the values of $\delta_j = 1$ if $\hat{b}_j \neq 0$ and $\delta_j = 0$ if $\hat{b}_j = 0$. This is the well-known Big-M formulation, e.g. see [13] frequently used in the development of MILP models. Provided the lower (L_j) and upper (U_j) bounds are chosen to be sufficiently large, a solution to the MILP will be obtained.

2.2 Efficient L_0- norm regularization

The MILP (6) – (10) is non-smooth and therefore not as easy to solve for as, say, a Linear Program (LP). One way to overcome this difficulty is to introduce slack variables into the problem. The non-smooth, MILP can be cast into the following equivalent smooth, constrained problem which is more amenable to solution,

$$J(\lambda) = \sum_{i=1}^{N} z_i + \lambda s \tag{11}$$

$$\sum_{j=1}^{r} \delta_j - s \leq 0 \tag{12}$$

$$s \in \mathbb{Z}_{\geq 0}, 0 \leq s \leq N$$

The additional constraints (12) ensure that the slack variable, *s*, (which is an integer and an additional decision variable for the MILP) provides a smooth penalty that represents the constraint violation. At the optimum, the slack variable, *s*, will be equal to the value of, $\sum_{j=1}^{r} \delta_j$ if the constraints are satisfied. This formularization has been successfully used in the process control literature e.g. see [14]-[16], for the development of constrained model predictive control algorithms. Essentially, it allows the MILP to 'soften', i.e. violate the constraints, if no alternative solution can be found thereby promoting a more efficient search.

2.3 Implementation of LASSO via MILP

If the binary variable constraint (9) is replaced by the weaker constraint that each of the δ_j belong to the interval [0 1], the NP-hard optimization problem may be transformed into a related problem that is solvable in polynomial time (Linear Programming rather than MILP). It is straightforward to verify that if the upper and lower bounds on the model parameters are specified as,

$$U_j(j = 1, \dots, r) = M \text{ and } L_j(j = 1, \dots, r) = -M$$

The optimal value of the 'relaxed' binary variables will be, $\delta_j = \hat{b}_j / M$ - see constraint (8). Therefore the relaxed MILP is equivalent to the LASSO (performing L_1 – norm regularization) where,

$$\left\|\hat{\mathbf{b}}\right\|_0 \approx \frac{1}{M}\left\|\hat{\mathbf{b}}\right\|_1$$

[17] and [18] have considered strategies to tighten the value of the upper bound to enhance the solution efficiency of a cardinality constrained quadratic optimisation strategy. However,

optimal determination of these bounds is itself a combinatorial problem (all subsets of models would have to be regressed to get the true upper bounds for each of the model parameters). Therefore, in the next section of the paper, an iterative approach is proposed where nonlinear function approximations to the L_0-norm are used to reweight the relaxed binary variables in order to implement cardinality constrained regression.

2.4 Approximating the L_0 – norm penalty and sequential linear programming

As the name suggests, sequential linear programming (SLP) is an iterative optimisation approach that is realized through linearization of any nonlinear terms (either in the cost function or constraints) around a current solution point. The solution of the resulting linear program (LP) is then used as a new point to solve the nonlinear problem and this is continued until a stopping criterion is met. To develop a SLP approach to L_0- norm regularization, a first order Taylor series approximation of the regularization penalty is considered (where 'k' represents an iteration index and $\boldsymbol{\delta}$ is a vector of the relaxed binary variables; non-negative functions of the model parameters), giving at the $(k+1)^{th}$ iteration,

$$P(\hat{\mathbf{b}}_{k+1}) \approx P(\hat{\mathbf{b}}_k) + \left.\frac{dP(\hat{\mathbf{b}})}{d\boldsymbol{\delta}}\right|_k (\boldsymbol{\delta}_{k+1} - \boldsymbol{\delta}_k)$$

As $P(\hat{\mathbf{b}}_k)$ and $\left.\frac{dP(\hat{\mathbf{b}})}{d\boldsymbol{\delta}}\right|_k \boldsymbol{\delta}_k$ are constants they will not affect the optimal solution at the $(k+1)^{th}$ iteration, therefore the objective function to be minimized is,

$$J_{k+1}(\lambda) = \sum_{i=1}^{N} z_{i,k+1} + \lambda \sum_{j=1}^{r} \frac{dP(\hat{\mathrm{b}}_{j,k})}{d\delta_{j,k}} \delta_{j,k+1} \tag{13}$$

If the penalty term is taken as $P(\hat{\mathbf{b}}) = \sum_{j=1}^{r} \delta_j$ minimization of (13) with respect to the constraints (7), (8), (10) and $\delta_j \in [0,1]$ is the relaxed MILP (the LASSO). Alternative realizations of (13) may be obtained through the use of nonlinear penalty functions that have been suggested in the literature as approximations to the L_0-norm. Consider, for example, the nonlinear approximation (4), defined in terms of the relaxed binary variables,

$$P(\hat{\mathbf{b}}) = \sum_{i=1}^{r} \log(\delta_j + \epsilon_1), \epsilon_1 > 0$$

Then (13) becomes,

$$J_{k+1}(\lambda) = \sum_{i=1}^{N} z_{i,k+1} + \lambda \sum_{j=1}^{r} \frac{\delta_{j,k+1}}{\delta_{j,k} + \epsilon_1} \tag{14}$$

The formula for the penalty term of this SLP approach to cardinality constrained regression using two alternative nonlinear approximations to the L_0 – norm suggested in the literature are provided in Table 1 (in the results the performance of each of these penalties is compared). While the mathematical structures differ, the mechanism used to ensure sparsity is the same. The effective regularization weight increases rapidly for small values of $\delta_{j,k}$, it is therefore better to set some $\delta_{j,k+1}$ to zero at the next iteration than obtain a compromise solution where a number of model parameters are non–zero.

A single iteration of the (13) using any of the penalty terms corresponds to the LASSO (provided the SLP is initialized appropriately, e.g. (14) could be initialized as, $k = 0, \delta_{j,0} + \epsilon = 1$). However, further iteration may proceed until a stopping criteria, e.g.

$|J_{k+1}(\lambda) - J_k(\lambda)| \leq \gamma$ (where γ is a small tolerance), has been reached. This iterative re-weighting of a penalty function has been indirectly used in reweighted L_1 penalized regression, see e.g. [19], [20] who demonstrated the effectiveness of the approach (applied in the area of compressive sensing addressing the recovery of sparse signals). [21] use the DC (difference of convex functions) programming framework to derive a variant of (14) through the consideration of nonlinear penalty terms. They subsequently solve the resulting nonlinear optimisation problem using an extension to the coordinate-wise LASSO optimisation approach of [22].

	$P(\hat{\mathbf{b}})$	$\frac{dP(\hat{\mathrm{b}}_{j,k})}{d\delta_{j,k}}$
Exponential [9]	$1 - e^{-\epsilon_2 \lvert b_j \rvert}$	$\epsilon_2 e^{-\epsilon_2 \lvert \delta_{j,k} \rvert}$
SELO [12]	$\frac{1}{\log(2)} \log\left(\frac{\lvert b_j \rvert}{\lvert b_j \rvert + \epsilon_3} + 1\right)$	$\frac{1}{\log(2)}\left(\frac{\epsilon_3}{(2\delta_{j,k} + \epsilon_3)(\delta_{j,k} + \epsilon_3)}\right)$

Table 1. Alternative formula for the linearized penalty term based on nonlinear penalties using an exponential function and the Seamless L_0 – or SELO. The constants ϵ_2 and ϵ_3 are 'tuning' parameters that affect the shape of the penalty.

2.5 Model structure selection

The choice of the optimal regularization parameter (λ) is an important issue and this may be achieved using a model validation strategy. Generally, cross validation strategies are used, determining the optimal λ by finding the minimum of the prediction error on a test (or validation) data set. However, choosing λ in this manner can be computationally intensive. Alternative approaches are to use information criteria such as AIC and BIC. It is known that AIC-based methods are not consistent for model selection as irrelevant model parameters tend to be selected e.g. see [23]. Therefore, in this work the BIC criterion is used which may be described by,

$$BIC = -2\,ln(lik) + ln(N)df$$

Where $'lik'$ is the maximum value of the likelihood function of the model, df is the number of parameters (degree of freedom) of the model. Using the LAD cost function the BIC cost function is,

$$BIC(\lambda) = N \ln(\|\boldsymbol{\varepsilon}\|_1/N) + ln(N)df \qquad (15)$$

An optimal model structure corresponds to the regularization parameter λ that minimizes (15). Therefore, the MILP may be minimized for a range of λ values in order to determine the regularization path (or landscape) and the corresponding values of (15) are calculated to determine the optimal model structure. For LASSO, [24] prove that the number of non-zero coefficients within the model is an unbiased estimator of the model degree of freedom, df; for our MILP strategy this is the sum of the binary variables associated with each model parameter. For the relaxed implementations of the MILP, the number of non-zero values may be heuristically determined.

Fortunately, unlike LASSO the determination of the optimal weighting for a L_0-norm constrained MILP does not require cross-validation. The optimal value of λ may be specified directly from the information criteria such as AIC, BIC etc. (each of which are known to be

optimal given certain assumptions about the data). For example, in using the BIC, $\lambda_{opt} = \log(N)$. This allows model parameter estimation and structure selection to be achieved in a single step for the MILP algorithms directly using the L_0 - norm, (11) and (12). In the results that follow we make the additional assumption that the SLP algorithms derived using the nonlinear approximation to the L_0-norm are sufficiently accurate to justify the direct specification of λ_{opt} using the BIC criteria.

3.0 Results

The first two examples presented in this section of the paper are used to compare the performance of the MILP using the cardinality constraint (as well as the implementation using the slack variable), the relaxed-MILP (LASSO) and the use of SLP to implement L_0 – norm regularization. The third example is an application to the development of a calibration model for prediction with Near Infrared (NIR) data. The data set comprises spectral intensities of 60 samples of gasoline at 401 wavelengths and their octane ratings. The data is available in MATLAB and used as a demonstration for Principal Component Regression (PCR) and Partial Least Squares (PLS). The final example considers the development of a model for prediction of chemical toxicity. The model is developed using the descriptors which characterize behavior (from the commercial DRAGON package). For all examples reported the MILP is solved using the function 'intlinprog' with default settings in MATLAB.

3.1 Example 1

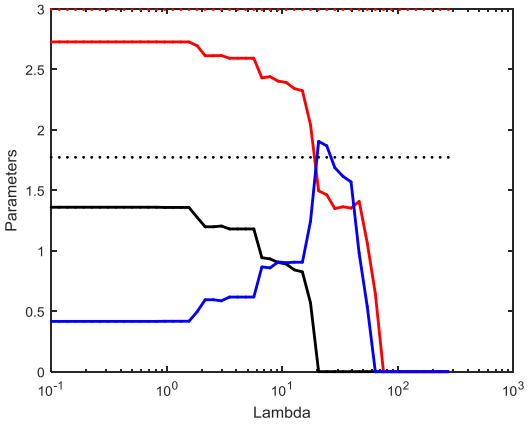
This example was originally presented by [24] to show conditions where LASSO does not choose the correct model. While this simulated example is a small problem, it is presented to demonstrate the characteristics of L_1 – norm and L_0 – norm regularization across an entire regularization path. Independent and identically distributed random variables x_1, x_2, e, σ were generated $(N = 100)$ with a mean of zero and a variance of one. An additional random variable is constructed as,

$$x_3 = \frac{2}{3}x_1 + \frac{2}{3}x_2 + \frac{1}{3}e$$

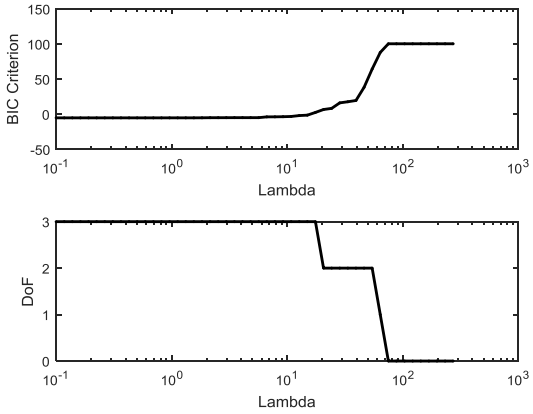
Giving the matrix of input data, $\mathbf{X} = [x_1 \quad x_2 \quad x_3]$. The model parameters are specified as, $\mathbf{b} = [2 \quad 3 \quad 0]^T$ and the output data generated using the linear model, $\boldsymbol{y} = \mathbf{X}\mathbf{b} + \boldsymbol{\sigma}$. The objective is to determine the true model structure and model parameters using the input ($\mathbf{X}$) and output ($\boldsymbol{y}$) data.

To perform model regularization, λ is increased from an initial value of 0.1 to a final value of λ_{max} over fifty log-spaced intervals. The value of λ_{max} is taken to be $\lambda_{max} = \text{sum}(\text{abs}(\boldsymbol{y}))$ which is the value of the LAD cost function when $\hat{\mathbf{b}} = \mathbf{0}$. For all experimental runs the upper and lower bounds on the parameter values were specified as, $(L_j, U_j) = (0, 3)$.

Fig. 1 demonstrates the model regularization using the LASS0 (implemented using the relaxed MILP). In Fig. 1a the transition of the model parameters may be observed as the regularization weight is increased. The dashed lines represent the best estimate of the true model parameters (assuming the model structure is known). It may be observed that the LASS0 produces biased parameter estimates and does not recover the true model structure (or parameters). In Fig. 1b the value of the BIC criterion as well as the number of model parameters as a function of the regularization weight are shown. The best model obtained using the LASS0 (with the minimum BIC criterion) corresponds to a regularization weight, $\lambda = 0.1179$ with estimated model parameters, $\hat{\mathbf{b}} = [1.3752 \quad 2.7265 \quad 0.4238]^T$.



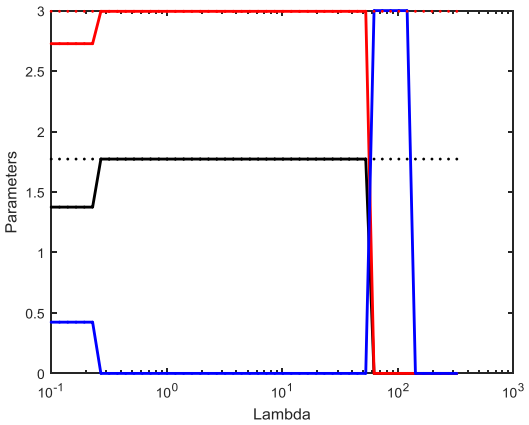
(a) Model regularisation using the LASSO



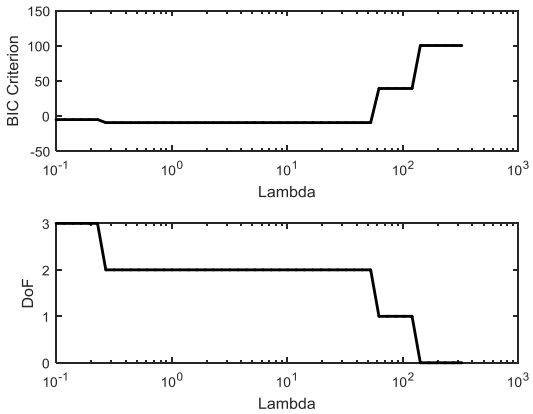
(b) BIC Criterion and the number of model parameters (DoF)

Fig. 1 Model regularization using the LASS0. (a) The estimated model parameter profiles as the regularization parameter is increased. The "true" (best estimate given the known model) parameters are shown as dashed lines. (b) The BIC criterion and the number of model parameters (DoF) as a function of the regularization parameter.

Fig. 2a shows the regularization landscape using the L_0 – norm penalty function. Here, the parameter estimates stay constant over a wide range of regularization weights and when sufficient regularization pressure is applied there is a shift in the parameter estimates (corresponding to a new model structure). Initially, with low values of λ there are three (biased) parameters within the model. At a value of $\lambda = 4.605$ (corresponding to the optimal BIC criterion weight) the estimated model parameters are, $\hat{\mathbf{b}} = [1.7724 \quad 2.9940 \quad 0]^T$; corresponding exactly to the parameter estimates obtained using the known model structure. As the value of λ is increased it may be observed that the correct model parameters are removed from the model, while the third parameter re-appears; this sub-optimal model structure having an increased value of the BIC criterion. At the end of the regularization process all model terms are set to zero as indicated in Fig. 2b.



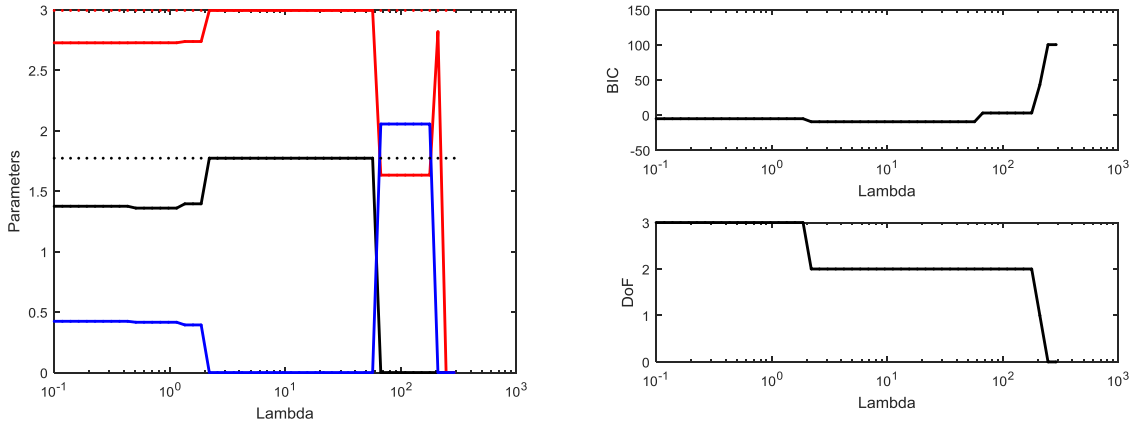
(a) Model regularisation using the L_0 norm



(b) BIC Criterion and the number of model parameters (DoF)

Fig. 2 Model regularization using the L_0 norm. (a) The estimated model parameter profiles as the regularization parameter is increased. The "true" (best estimate given the known model) parameters are shown as dashed lines. (b) The BIC criterion and the number of model parameters (DoF) as a function of the regularization parameter.

Fig. 3 shows the performance of the SLP approach (the results for the exponential penalty function are shown) to L_0-norm regularization. The approximate penalty function was specified with a value of ϵ_3=30 and the convergence criteria for the iterations, $|J_{k+1}(\lambda) - J_k(\lambda)| \leq 1\text{x}10^{-5}$. As with the L_0-norm at small values of λ there are three (biased) parameters within the model. At a value of $\lambda = 4.605$ (corresponding to the optimal BIC criterion weight) the estimated model parameters are, $\hat{\mathbf{b}} = [1.7724 \quad 2.9940 \quad 0]^T$; corresponding exactly to the parameter estimates obtained using the known model structure. The average number of iterations required in order for the algorithm to converge was four. Identical results were obtained for the SLP implemented using the SELO (using ϵ_2=0.01) and identical stopping criteria. However, while the SLP implemented using the logarithm penalty (using ϵ_3=0.05) correctly identified the correct model structure the parameter estimates, $\hat{\mathbf{b}} = [1.6566 \quad 3.0286 \quad 0]^T$ were biased.



(a) Model regularisation using the L_0 norm approximation (using SLP)

(b) BIC Criterion and the number of model parameters (DoF)

Fig. 3 Model regularization using the L_0 norm approximation (using SLP and the exponential penalty function). (a) The estimated model parameter profiles as the regularization parameter is increased. The "true" (best estimate given the known model) parameters are shown as dashed lines. (b) The BIC criterion and the number of model parameters (DoF) as a function of the regularization parameter.

3.2 Example 2

The following linear model is simulated,

$$\boldsymbol{y} = \mathbf{X}\mathbf{b} + \boldsymbol{\sigma}$$

Using the parameter vector (the values of $q = 0, 40, 90, 140, 190$ give a linear regression problem with r = 10, 50, 100, 150 and 200 variables),

$$\mathbf{b} = [1.5 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 1.0 \quad -1.0 \quad 0 \quad 0 \quad \mathbf{0}(1\text{x}\,q)]^{\text{T}} \qquad (18)$$

Input data (N = 200) was generated from a Gaussian distribution with a mean of zero and the covariance structure, $\sum(i,j) = 0.3^{|i-j|}$ (arbitrarily chosen to reflect the fact that in many practical examples of sparse regression the input variables are correlated). Noise ($\boldsymbol{\sigma}$) with a mean of zero and a standard deviation defined to give an output-signal to noise ratio of 10% was added to the process output signal. For all experimental runs the upper and lower bounds on the parameter values were specified as, $(L_j, U_j) = (-2, 2)$. To solve the SLP the same penalty function, tolerances and stopping criteria were used as in example 1. The results of one hundred Monte Carlo simulations are presented in Table 2. To perform model regularization, for the LASSO, λ was increased from an initial value of 0.1 to a final value of λ_{max} over fifty

log-spaced intervals. The value of λ_{max} is taken to be $\lambda_{max} = \text{sum}(\text{abs}(\boldsymbol{y}))$. For the cardinality constrained algorithms, the value of $\lambda_{opt} = 5.3$.

r		MAE	$\|\hat{b} - b\|_2^2$	CM (%)	FP (%)	FN (%)	run-time
10	**LASSO**	0.1539 $\pm$ 0.009	0.0039 $\pm$ 0.003	52	48	0	1.0
	SLP	0.1512 $\pm$ 0.008	$9.97\text{x}10^{-4}$ $\pm$ $8.56\text{x}10^{-4}$	100	0	0	0.22
	MILP	0.1514 $\pm$ 0.008	0.001 $\pm$ $9.55\text{x}10^{-4}$	99	1	0	0.23
	MILP (slack)	0.1517 $\pm$ 0.008	$7.83\text{x}10^{-4}$ $\pm$ $5.91\text{x}10^{-4}$	100	0	0	0.20
50	**LASSO**	0.1607 $\pm$ 0.0115	0.0083 $\pm$ 0.005	39	61	0	2.0
	SLP	0.1506 $\pm$ 0.007	$9.54\text{x}10^{-4}$ $\pm$ $7.78\text{x}10^{-4}$	100	0	0	0.32
	MILP	0.1514 $\pm$ 0.008	$9.954\text{x}10^{-4}$ $\pm$ $7.61\text{x}10^{-4}$	100	0	0	2.23
	MILP (slack)	0.1501 $\pm$ 0.008	$8.42\text{x}10^{-4}$ $\pm$ $6.82\text{x}10^{-4}$	100	0	0	0.36
100	**LASSO**	0.1615 $\pm$ 0.011	0.0114 $\pm$ 0.005	24	76	0	5.02
	SLP	0.1509 $\pm$ 0.0072	0.0011 $\pm$ $9.51\text{x}10^{-4}$	98	2	0	0.52
	MILP	-	-	-	-	-	305*
	MILP (slack)	0.1499 $\pm$ 0.007	0.001 $\pm$ $9.64\text{x}10^{-4}$	100	0	0	2.61
150	**LASSO**	0.1637 $\pm$ 0.0115	0.0130 $\pm$ 0.005	32	68	0	6.91
	SLP	0.1504 $\pm$ 0.00886	$9.07\text{x}10^{-4}$ $\pm$ $7.79\text{x}10^{-4}$	94	6	0	0.77
	MILP	-	-	-	-	-	4778*
	MILP (slack)	0.1494 $\pm$ 0.0076	$8.59\text{x}10^{-4}$ $\pm$ $6.79\text{x}10^{-4}$	97	3	-	17.92
200	**LASSO**	0.1249 $\pm$ 0.0713	1.2396 $\pm$ 2.907	13	87	-	10.77
	SLP	0.1506 $\pm$ 0.008	0.0012 $\pm$ 0.0012	90	10	-	1.2
	MILP	-	-	-	-	-	13,977*
	MILP (slack)	-	-	-	-	-	244*

Table 2. Comparison of the performance of the MILP relaxed-MILP (LASSO) and the use of SLP to implement relaxed L_0 – norm regularization. MAE is the mean absolute error between the actual outputs and the outputs of the model generated by the MILP over all the Monte Carlo simulations (the MAE is used as the cost function for the MILP is defined in terms of absolute error), $\|\hat{b} - b\|_2^2$ is the 2- Norm between the identified model parameters and the true model parameters used to generate the data. CM indicates the percentage of correct model structures identified. A false positive (FP) model is defined as a model that includes at least one additional parameter when compared to the true model. A false negative (FN) model is defined as a model that has at least one missing parameter when compared to the true model. Relative run time is defined as being relative to the time it took to solve the LASSO with *r* = 10 variables (*estimated relative run-time from a single Monte Carlo simulation).

For the MILP (slack), which is the MILP implemented using the slack variable, the true parameter vector is discovered in 100% of the Monte Carlo simulations for *r* = 10, 50, 100 variables and 97% of the models had the correct structure for *r* = 150. Relative run time (all relative to the time it took to solve the LASSO with *r* = 10 variables across an entire

regularization path using fifty log-spaced values of λ) of the MILP (slack) increased for r = 150, 200 variables (therefore the result is only reported for one simulation where the correct model structure was found). In comparison the direct implementation of the L_0-norm regularization strategy (referred to as MILP in the tables) using (6)-(10) however, the relative run-times are significantly reduced. It may be observed in Table 2 that for problems involving a small number of variables, $r \leq 100$ the MILP (slack) is more efficient than the LASSO. This is because a single value of the regularization parameter, $\lambda_{opt} = 5.3$ is used whereas for the LASSO is performed across the entire regularization landscape. The SLP was implemented using equation (14) and the performance (in terms of model structure and parameter estimation) is similar to the MILP (slack). However, as the number of model parameters increase to r = 150, 200 the relative run-time of the SLP is significantly lower than the MILP (slack).

In all the simulations, LASSO performance (in terms of obtaining the correct model structure) is significantly inferior to the L_0-norm regularization strategy tending to identify false positive models (with the percentage of false positive models increasing as the number of potential parameters increase). In Table 2 it may also be noted that the consistent identification of the correct model structure using the L_0- norm strategy provides consistent values for the MAE and 2-Norm of the parameter vector. While for the LASSO, the MAE and 2-Norm of the parameter vector increase with an increase in the number of potential model parameters (a result of the increase in the number of false positive models).

To test the performance of the various algorithms using data with alternative covariance structures, the Monte Carlo simulations were repeated (the number of simulations was again 100). The true model was again taken to be (18) with r = 100 potential regression variables. Three covariance structures $\sum(i,j) = c^{|i-j|}$ with c = 0, 0.5 and 0.8 respectively were investigated. Noise ($\boldsymbol{\sigma}$) with a mean of zero and a standard deviation defined to give an output-signal to noise ratio of 10% was added to the process output signal.

c		MAE	$\|\hat{b} - b\|_2^2$	CM (%)	FP (%)	FN (%)	run-time
0	**SLP (log)**	0.1633 $\pm$ 0.0084	0.0011 $\pm$ 8.33x10^{-4}	96	4	0	1.2
	SLP (exp)	0.1623 $\pm$ 0.0079	0.0011 $\pm$ 9.71x10^{-4}	99	1	0	1.0
	SLP (SELO)	0.1625 $\pm$ 0.0096	0.0011 $\pm$ 8.99x10^{-4}	100	0	0	1.16
	MILP (slack)	0.1627 $\pm$ 0.0089	0.0011 $\pm$ 8.32x10^{-4}	98	2	0	7.68
0.5	**SLP (log)**	0.1427 $\pm$ 0.0082	0.0011 $\pm$ 0.001	96	4	0	1.2
	SLP (exp)	0.1425 $\pm$ 0.0077	9.1x10^{-4} $\pm$ 7.1x10^{-4}	100	1	0	1.0
	SLP (SELO)	0.1431 $\pm$ 0.0079	0.001 $\pm$ 9.17x10^{-4}	100	0	0	1.15
	MILP (slack)	0.1426 $\pm$ 0.0088	0.0011 $\pm$ 0.0013	99	1	0	4.96
0.8	**SLP (log)**	0.1323 $\pm$ 0.0062	0.0034 $\pm$ 0.0032	99	1	0	1.25
	SLP (exp)	0.1327 $\pm$ 0.0078	0.0013 $\pm$ 0.0014	100	0	0	1.0
	SLP (SELO)	0.1317 $\pm$ 0.0075	0.0019 $\pm$ 0.0022	100	0	0	1.2
	MILP (slack)	0.1331 $\pm$ 0.0071	0.0013 $\pm$ 0.0012	99	1	0	3.69

Table 3. The performance of the MILP (slack) and the SLP algorithms for three alternative covariance structures (r = 100 potential regression variables). The relative run-time is reported with respect to the most efficient (fastest) algorithm for each correlation structure.

Table 3 compares the performance of the MILP (slack) with each of the SLP implementations. The run-time is reported with respect to the most efficient algorithm for each covariance structure. There is only a small difference in run-time for the SLP algorithms (caused by slight variations in the number of iterations required to converge to the optimal solution). An interesting observation with the MILP (slack) is that the relative run-time improved as the correlation between the variables increased. In terms of model parameter and structure identification, for each covariance structure, a high percentage of correctly identified models are obtained.

3.3 Regression of spectra data

The data set (available with MATLAB) consists of octane number (octane) and NIR spectra (NIR) of 60 gasoline samples [26]. Each NIR spectrum consists of 401 diffuse reflectance measurements from 900 to 1700 nm. The objective is to predict the octane number. Since, $r \gg N$ the problem may be categorized as a high dimensional data set and methods such as PCR and PLS regressions have been shown to be successful in developing models using this data. In order to implement the MILP on this data set, it is possible to reduce the dimensionality of the parameter regression problem, if it is assumed that the model parameters, $\hat{\mathbf{b}}$ are a linear combination of $\mathbf{X}$ such that, $\hat{\mathbf{b}} = \mathbf{X}^{\mathbf{T}}\boldsymbol{w}$ giving

$$\boldsymbol{y} = \mathbf{X}\mathbf{X}^{\mathbf{T}}\boldsymbol{w} + \boldsymbol{\sigma}$$

The matrix $\mathbf{X}\mathbf{X}^{\mathbf{T}}$ (N x N) is referred to as a linear kernel in support vector regression, e.g. see [27]. The advantage when $r \gg N$ is that the number of unknown model coefficients, $\boldsymbol{w}$ (r x 1) to be estimated is reduced. Model regularization is therefore performed to determine the optimal subset of $\boldsymbol{w}$, and as with PCR and PLS the optimal predictor is a function of all the dependent variables.

To identify a prediction model using the MILP (slack) and SLP the upper and lower bounds on the parameter values were specified as, $(L_j, U_j) = (-0.5, 0.5)$. To solve the SLP the same penalty function, tolerances and stopping criteria were used as in example 1. The input – output data was scaled to be in the range [-1, 1]. The data samples were then partitioned into a training data set and a testing (validation data set), with 50 samples in the training data and 10 in the validation (chosen randomly). A value of $\lambda_{opt} = 3.91$ was used in the experimental runs. In order to compare the performance of the algorithms, models were also developed using PCR and PLS.

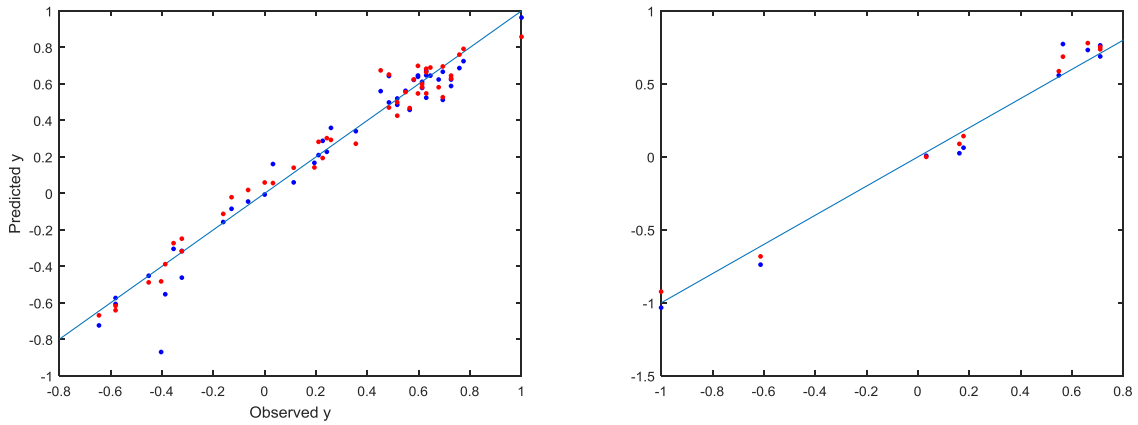
Fig. 4 shows the prediction performance of the model obtained using the MILP (slack) and compares this performance to the best model obtained using PLS. In Table 4 a summary of the training and testing MAE as well as the optimal number of model terms are presented for the MILP (slack), SLP (the penalty function used was the logarithm penalty using the same settings as example 1), PCR and PLS.

It may be observed that the performance of each algorithm is comparable. The PLS model provides the lowest MAE (on the training and validation data set), the next best performance is that obtained using PCR. The performance of the MILP (slack) and SLP (log) are slightly inferior, however as can be seen in Fig. 4 this is not significant. The one "outlier", which can be observed for the MILP (slack) in Fig. 4a, might be due to the minimization of the absolute error with the proposed approaches, which is known to be more robust towards outliers. However, both, the MILP (slack) and SLP (log), use only three/five inputs (which are the

support vectors that constitute the basis of support vector machines), whereas the PLS and PCR use all inputs. It could be argued therefore that the model structures identified by MILP (slack) and SLP (log) are more parsimonious than those of the PCR and PLS.

	MAE (Training)	MAE (Validation)	DoF
MILP (slack)	0.0632	0.0799	3*
SLP (log)	0.0778	0.0819	5*
PCR	0.0612	0.0688	4**
PLS	0.0613	0.0635	3**

Table 4 MAE (training and validation) and the model DoF for each of the modelling techniques tested. * In case of MILP and SLP the DoF are the number of support vectors. ** In case of PCR and PLS, the DoF are the number of latent variables.



(a) Observed y versus predicted y (training data). Blue dots slack MILP the red PLS

(b) Observed y versus predicted y (testing data) Blue dots slack MILP the red PLS

Fig. 4 A comparison of the observed and predicted response of the models developed using the MILP (slack) and PLS. The data consists of octane number (octane) and NIR spectra (NIR) of 60 gasoline samples [26].

3.4 QSAR

QSAR (Quantitative Structure Activity Relationships) is a well established technique for deriving structure property relationships for chemical compounds that can be used to predict the properties of novel chemical structures. Chemical compounds can be represented by a large number of computed numerical values, called “descriptors”, each of which in some way characterize the structure of the compound. The idea of QSAR is to build empirical or semi-empirical models that relate the descriptors of a compound to some physical, chemical or biological property. Software packages are available to compute descriptor values for compounds with a known structure. Many of these are commercial products (e.g. DRAGON) but there are also free/open source packages (e.g. the Chemical Development Kit (CDK; [28]).

QSAR uses a data set of known chemical compounds and a measured endpoint for each compound. The measured endpoint is the property of interest. Typical properties of interest are those related to pharmaceutical drug development. These include biological activities representing the ability of a compound to perform its desired function (e.g. IC50, the concentration of a compound required to inhibit a particular biological or biochemical function by half) and the ADME properties (adsorption, distribution, metabolism and excretion) which characterize the behavior of a pharmaceutical drug compound within the organism.

The prediction of chemical toxicity is another chemical property that is of vital importance in both pharmaceutical drug development and managing the environmental risk of chemical compounds. In the latter case there are legal regulatory structures (e.g. the REACH

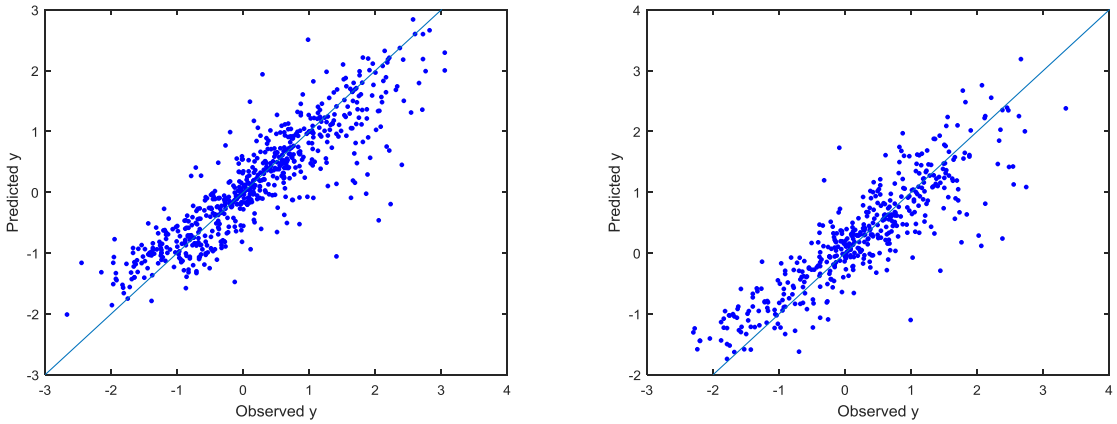
regulations in the European Union - EC 1907/2006) that specify that QSAR models should play a part in managing this risk in order to reduce the costs of experimental toxicity measurement. One method for experimentally evaluating chemical toxicity is the measurement of the growth inhibition of ciliated protozoan T. pyriformis. There are freely available aquatic toxicity data for more than 1000 compounds, due to the efforts of [29]. [30] have used this to compile a data set of 1093 unique compounds and have developed a number of predictive QSAR models using various descriptor packages and modelling methodologies. Here, we demonstrate the development of a linear predictive model of chemical toxicity using this data set (using the descriptors from the commercial DRAGON package) and the results compared with those published in [30] and [31].

The T. pyriformis toxicity values (i.e. the response y data) are measured as the logarithm of the 50% growth inhibition concentration $\log(IGC50^{-1})$. The data available for training QSAR models contains 644 compounds and 449 compounds are used as an external test/validation data set to verify the predictive ability of the models. For each compound 1664 DRAGON descriptor values are used as the predictor data (i.e. the input X data contains 1664 input variables) - compound structures, toxicity and descriptor values are, at time of writing, available from the EU CADASTER website at http://www.cadaster.eu/node/65.

As with example 3, it is assumed that the model parameters, $\hat{\mathbf{b}}$ are a linear combination of $\mathbf{X}$ such that, $\hat{\mathbf{b}} = \mathbf{X}^{\mathbf{T}}\boldsymbol{w}$ giving

$$\boldsymbol{y} = \mathbf{X}\mathbf{X}^{\mathbf{T}}\boldsymbol{w} + \boldsymbol{\sigma}$$

Model regularization is therefore performed to determine the optimal subset of $\boldsymbol{w}$. To identify a prediction model the SLP algorithms were used with the upper and lower bounds on the parameter values were specified as, $(L_j, U_j) = (-0.5, 0.5)$. To solve the SLP the same penalty function, tolerances and stopping criteria were used as in example 1. The input – output data was scaled to be in the range [-1, 1]. A value of $\lambda_{opt} = 3.91$ was used in the experimental runs. Fig. 5 shows the prediction performance of the model obtained using the SLP (log) on the training and validation data set.



(a) Observed y versus predicted y (training data)

(b) Observed y versus predicted y (testing data)

Fig. 5 A comparison of the observed and predicted response of the model developed using the SLP (log) and the training and testing (validation) data. The data available for training the QSAR model contained 644 compounds and 449 compounds are used as an external test/validation data set to verify the predictive ability of the models.

[30] report their results in terms of MAE (mean absolute error) for two test sets - referred to in their paper as Validation set 1 (339 compounds) and Validation set 2 (110 compounds) that

comprise the whole test set used here. In terms of MAE, the best model obtained using the SLP (log) has an MAE(training) = 0.3506, and MAE(test) = 0.3729. This compares to our earlier results [31], that used a genetic programming (GP) to develop a nonlinear regression model minimizing 'goodness of fit' to the data of MAE(training) = 0.3292 and MAE(test) = 0.3518. The identified model is of comparable accuracy to that obtained using the computationally expensive global search technique of GP and a compact, linear model was obtained containing 22 variables automatically selected from 664 possible variables.

In [30] the results of a number of individual models (and ensemble models) are reported, built using various descriptor packages and modelling techniques. Some of these models consider the "applicability domain" (AD) of the compounds (i.e. whether the compounds lie in the region of descriptor space deemed to be suitable for generating a prediction) whereas others do not employ AD considerations. In general, models that consider AD give more accurate predictions but only the results of the non AD models using the DRAGON descriptors are repeated here.

The first DRAGON descriptor based model is a support vector machine (SVM) regression that yields MAE(Validation set 1) = 0.37 and MAE(Validation set 2) = 0.42. This corresponds to an MAE(test) = 0.38. The second DRAGON based model is a k- nearest neighbor (k-NN) approach that achieves MAE(Validation set 1) = 0.29, MAE(Validation set 2) = 0.43 corresponding to MAE(test) = 0.32. Hence it can be seen that the developed model has achieved predictive performance of the order of the current state of the art empirical modelling methodologies while ensuring that a low complexity structure is obtained.

4.0 Discussion and conclusions

Several works have shown that alternative penalties to the L_1-norm can improve sparse regression. In this paper MILP has been used as a framework for solving L_0-norm regularized regression. For problems where $r \sim 100 - 150$ variables it has been demonstrated that this may be achieved without approximation. For regression problems of higher dimension, approximation of the L_0-norm by nonlinear functions allows sparse regression using SLP. The results presented in the paper demonstrate the MILP (slack) and the SLP algorithms efficiently solved sparse regularization problems (both in terms of computational overhead and accuracy of the resulting model structures and parameters). To the best of our knowledge, this is the first time that L_0-norm applications (without approximation or relaxation) with input dimension >50 have been shown to be efficiently and accurately solved.

The performance of the SLP algorithms will be dependent on their respective tuning parameters, $\epsilon_i (i = 1, \dots, 3)$. In this work, fixed values of these tuning parameters were used in order to demonstrate the capability of the SLP approach to parsimonious model development. In principle when implementing these methods, the values of these parameters may be fine-tuned, e.g. via an internal optimization loop. It would therefore be of interest to assess SLP algorithm performance improvements when an internal optimisation loop is used to define the optimal values of the tuning parameters.

Motivated by the elastic net [32] and the adaptive elastic net [33], it may be possible to also consider a mixed penalty involving the L_0-norm and the L_2-norm. The elastic net and the adaptive elastic net have been demonstrated to outperform sparse regression methods that do not involve an L_2 penalty in a number of settings. An assessment of the relative merits of the combined penalty would therefore be an interesting direction of research. For the non-relaxed implementation of the algorithms discussed in this paper this would necessitate the use of mixed integer nonlinear programming (MINLP). For the relaxed variants of the MILP,

implemented using SLP, this would merely involve the use of a modified nonlinear penalty that may be linearized as discussed in section 2.4 of this paper.

Finally, while linear regression techniques have been considered in this paper, it would also be interesting to consider the use of MILP for the development of nonlinear models. This could be achieved using for example, polynomial or Gaussian kernels. This would allow the benefits of L_0 – norm regularization (without approximation) to be studied within the wider applications of support vector machines.

5.0 References

[1] Filzmoser, P., Gschwandtner, M. and Todorov, V. (2012) Review of sparse methods in regression and classification with application to chemometrics. Journal of Chemometrics 26(3-4), 42–51.

[2] Akaike, H. (1974). A new look at the statistical model identification. IEEE T. Automat. Contr. 19, 716-723.

[3] Schwarz, G. (1978). Estimating the dimension of a model. Ann. Statist. 6, 461-464.

[4] Tibshirani, R. (1994) Regression shrinkage and selection via the lasso. Technical report, University of Toronto.

[5] Zou, H. (2006) The adaptive lasso and its oracle properties, Journal of the American Statistical Association, 101, 1418-1429

[6] Efron, B., Hastie, T., Johnstone, I. and Tibshirani, R. (2004) Least angle regression, The Annals of Statistics, 32, 2, 407–499

[7] Meinshausen, N. (2007) Relaxed Lasso, Computational Statistics & Data Analysis 52, 374 – 393

[8] Weston, J., Elisseef, A., Schölkopf, B. and Tipping, M. (2003) Use of the zero-norm with linear models and kernel methods, J. Mach. Learn. Res. 3, 1439–1461.

[9] Bradley, P.S. and Mangasarian, O.L. (1998) Feature selection via concave minimization and support vector machines. In Proc. 13th ICML, pages 82–90, San Francisco, CA.

[10] Fan, J. and Li, R. (2001). Variable selection via nonconcave penalized likelihood and its oracle properties. J. Amer. Statist. Assoc. 96, 1348-1361.

[11] Frank, I. E. and Friedman, J. H. (1993). A statistical view of some chemometrics regression tools. Technometrics, 35, 109–148.

[12] Dicker, L., Huang, B. and Lin, X. (2013) Variable selection and estimation with the seamless L0 penalty, Statistica Sinica, 23, 929-962

[13] Griva, I., Nash, S.G. and Sofer, A. (2009) Linear and Nonlinear Optimization, Society for Industrial and Applied Mathematics, Philadelphia.

[14] Oliveira, N.M.C and Biegler, L.T. (1994) Constraint handling and stability properties of model predictive control, American Institute of Chemical Engineers, vol. 40, pp. 1138–1155

[15] Kerrigan, E. and Maciejowski, J. (2000) Soft constraints and exact penalty functions in model predictive control, in Control 2000 Conference, Cambridge

[16] Richards, A. (2013) Fast Model Predictive Control with Soft Constraints, European Control Conference (ECC)

[17] Bertsimas, D. and Shioda, R. (2009) Algorithm for Cardinality-Constrained Quadratic Optimization, Computational Optimization and Applications, 43, 1, –22.

[18] Konno, H. and Yamamoto, R. (2009) Choosing the Best Set of Variables in Regression Analysis Using Integer Programming, Journal of Global Optimization, 44, 2, 272–282.

[19] Candes, E.J., Wakin, M.B. and Boyd, S.P. (2008) Enhancing sparsity by reweighted L1 minimization, J. Fourier Anal. Appl., 877–905.

[20] Chartrand, R. and Yin, W. (2008) Iteratively reweighted algorithms for compressive sensing, in Proceedings of International Conference on Acoustics, Speech, and Signal Processing (ICASSP'08), 3869–3872.
[21] Gasso, G., Rakotomamonjy, A. and Canu, S. (2009) Recovering sparse signals with a certain family of non-convex penalties and DC programming, IEEE Trans. Signal Processing, 57, 12, 4686-4698
[22] Friedman, J., Hastie, T., Hofling, H. and Tibshirani, R. (2007) Pathwise coordinate optimization, The Annals of Applied Statistics , 1, 2, 302–332
[23] Shao, J. (1993). Linear model selection by cross-validation. J. Amer. Statist. Assoc. 88, 486-494.
[24] Zou, H. Hastie, T. and Tibshirani, R. (2007) On the "degrees of freedom" of the lasso Ann. Statist. 35, 5, 2173-2192
[25] Zhao, P. and Yu, B. (2006) On model selection consistency of lasso, Journal of Machine Learning Research, 7, 2541–2563
[26] Kalivas, J.H. (1997) Two Data Sets of Near Infrared Spectra, Chemometrics and Intelligent Laboratory Systems, 37, 255-259.
[27] Vapnik, V.N., (2000) The nature of statistical learning theory, second edition, Springer-Verlag, New York.
[28] Steinbeck, C., Han, Y., Kuhn, S., Horlacher, O., Luttmann , E. and Willighagen, E. (2003) The Chemistry Development Kit (CDK): An open-source java library for chemo- and bioinformatics, J. Chem. Inf. Comput. Sci., 43, 493 – 500.
[29] Schultz T.W., Yarbrough, J.W. and Woldemeskel, M. (2005) Toxicity to Tetrahymena and abiotic thiol reactivity of aromatic isothiocyanates, Cell Biol. Toxicol., 21, 181-189
[30] Zhu, H., Tropsha, A., Fourches, D., Varnek, A., Papa, E., Gramatica, P., Oberg, T., Dao, P., Cherkasov, A. and Tetko, I.V. (2008) Combinatorial QSAR modeling of chemical toxicants tested against Tetrahymena pyriformis, J. Chem. Inf. Model.,48, 766 -784
[31] Searson, D., Leahy, D.E. and Willis, M.J. (2010) GPTIPS:An open source genetic programming toolbox for multigene symbolic regression, International MultiConference of Engineers and Computer Scientists (IMEC 2010), Vol I., Hong Kong.
[32] Zou, H. and Hastie, T. (2005). Regularization and Variable Selection via the Elastic Net, Journal of the Royal Statistical Society, Series B: 301–320.
[33] Zou, H. and Zhang, H. H. (2009). On the adaptive elastic-net with a diverging number of parameters. The Annals of Statistics, 37,1733–1751.